STATIONARITY AND SUPERLINEAR CONVERGENCE OF
AN ALGORITHM FOR UNIVARIATE LOCALLY
LIPSCHITZ CONSTRAINED MINIMIZATION

Robert MIFFLIN*
Department of Pure and Applied Mathematics, Washington State University, Pullman,
WA 99164-2930, USA

Received 16 August 1982
Revised manuscript received 18 March 1983

This paper introduces an algorithm for minimizing a single-variable locally Lipschitz function
subject to a like function being nonpositive. The method combines polyhedral and quadratic
approximation, a new type of penalty technique and a safeguard in such a way as to give
convergence to a stationary point. The convergence is shown to be superlinear under somewhat
stronger assumptions that allow both nonsmooth and nonconvex cases. The algorithm can be an
effective subroutine for solving line search subproblems called for by multivariable optimization
algorithms.

Key words: Nonsmooth Optimization, Nondifferentiable Programming, Constrained Minimization,
Locally Lipschitz Functions, Line Search.

1. Introduction

This paper introduces a rapidly convergent algorithm for solving the single-variable
constrained optimization problem of minimizing f on $C \subset R$ where $C = \{x \in R: c(x) \leq 0\}$ and $c$ and $f$ are locally Lipschitz functions defined on $R$ and a
convex set containing $C$, respectively. The method is an extension of one for the
convex case given in [7] which was based, in part, on one for the unconstrained
convex case given in [4]. The method efficiently combines quadratic and polyhedral
approximation (modified for negative curvature), the new penalty technique of [7]
and a safeguard to give convergence to a stationary point. This convergence result
of Section 3 does not require any extra assumption such as upper-semidifferentiability [1]. In Section 4 the convergence is shown to be superlinear for piecewise
semismooth functions where each piece is $C^2$ or convex. The algorithm may be
used as a line search subroutine for higher dimensional optimization algorithms,
because of its ability to deal with constraints, as described in the last section of
this paper. A listing of a BASIC language implementation of the algorithm is given
in [5].

* Research sponsored, in part, by the Air Force Office of Scientific Research, Air Force Systems
Command, USAF, under Grant Number AFOSR-83-0210. The U.S. Government is authorized to
reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.
For a locally Lipschitz function $F$ on a convex subset of $\mathbb{R}^n$ let $\partial F$ denote the set of subgradients (called generalized gradients in Clarke [2]) of $F$, i.e. $\partial F(x)$ is the convex hull (denoted $\text{conv}$) of all limits of sequences of the form

$$\{\nabla F(x_k): \{x_k\} \to x \text{ and } F \text{ is differentiable at each } x_k\}.$$  

If $F$ is convex $\partial F$ equals the subdifferential [8], i.e.

$$g \in \partial F(x) \text{ if and only if } F(y) \geq F(x) + g \cdot (y-x) \text{ for all } y.$$  

If $F$ is continuously differentiable (denoted $C^1$) $\partial F$ equals the ordinary gradient $\{\nabla F\}$. Furthermore, for many other functions $F$, such as those pieced together from $C^1$ functions, it is possible to determine $\partial F$ or at least to give one element of $\partial F(x)$ at each $x$. For example, if $F(x) = \max[\{F_1(x), F_2(x), \ldots, F_m(x)\}]$ where each $F_i$ is $C^1$, then $\partial F(x) = \text{conv}\{\nabla F_i(x): F_i(x) = F(x)\}$. Such a function may be given to us implicitly, i.e. for each $x$ we are given only $F(x)$ and $\nabla F_i(x)$ for one $i$ such that $F_i(x) = F(x)$.

Useful analytic properties of a locally Lipschitz function $F$ are that the point-to-convex set mapping $\partial F(\cdot)$ is upper semicontinuous and locally bounded and that the following mean value result holds [3]:

There exists an $\xi$ between $x$ and $y$ and a $\hat{g} \in \partial F(\xi)$ such that

$$F(x) - F(y) = \hat{g} \cdot (x-y).$$

$C$ is called the set of feasible points. We suppose that it is relatively easy to find a feasible point, for example, via unconstrained minimization of $c$. A point $x^* \in C$ is called stationary for $f$ on $C$ if either

$$c(x^*) < 0 \quad \text{and} \quad 0 \in \partial f(x^*)$$

or

$$c(x^*) = 0 \quad \text{and} \quad 0 \in \text{conv}(\partial f(x^*) \cup \partial c(x^*)), \quad \text{because this condition is necessary for } x^* \in C \text{ to minimize } f \text{ on } C. \text{ This condition is sufficient for } x^* \in C \text{ to be a constrained minimizer if } c \text{ and } f \text{ are semiconvex [6] and } 0 \notin \partial c(x^*) \text{ when } c(x^*) = 0.$$  

In addition to a starting feasible point, the algorithm only requires the value and one subgradient of $f$ at each feasible point and the value and one subgradient of $c$ at each infeasible point. Let the corresponding subgradient function be denoted by $g$, i.e.

$$g(x) = \begin{cases} \partial f(x) & \text{if } x \in C, \\ \partial c(x) & \text{if } x \notin C. \end{cases}$$

In general, $g$ is discontinuous, which is precisely the difficulty of nonsmooth optimization problems. Note that the algorithm can be applied to problems where $f$ is unknown outside of $C$ and/or $c$ is unspecified (other than being nonpositive) on $C$. 